

2 $A \times B + C \times D$

Load A

Multiply B

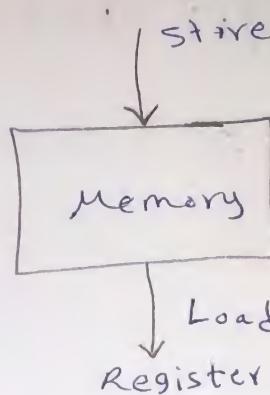
store Result

Load C

Multiply D

ADD Result

store result



Result

$$A \times B$$

Accumulator

$$C \times D$$

$$A \times B + C \times D$$

3

For ($J = n - 1$; $J \geq 0$; $J--$)

{

For ($K = J - 1$; $K \geq 0$; $K--$)

{

if ($List[K] \geq List^{[J]}$)

{

1

{
temp = List [K];

List [K] = List [J];

List [J] = temp

}

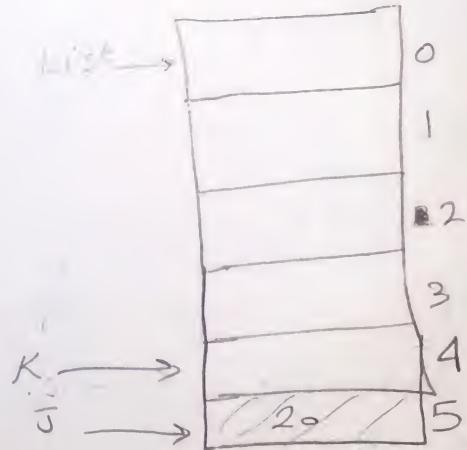
{

{

Note

J ← ثابت

K ← يتحرك للأعلى ويمرى هل
القيمة التي يشير إليها أكبر من
K ولا \rightarrow وعلى أساس ذلك هي بعد
الترتيب.



~~بعد الـ 5~~

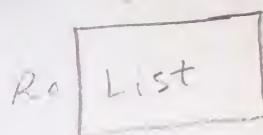
بعد الـ 5 ننتدار إلى (Loop) (1008)

جديدة فنقوم بتحريك J في مكان آخر ونثبيها.
ويحرك K كما سمع.

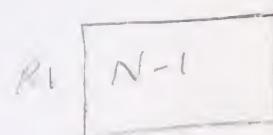
السؤال ١٣) مرتبة (insertion sort) دوستال د ← للسؤال عاليز برامج (assembly) (Code)

Sol

Move #List, R₀



Move N, R₁

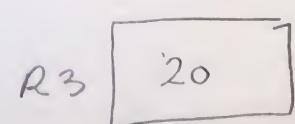


Subtract #1, R₁



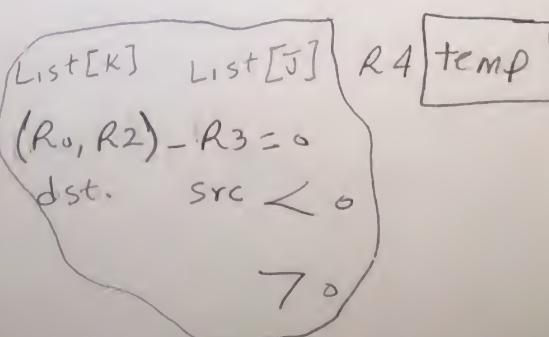
^{out} Move R₁, R₂

Subtract #1, R₂



Movebyte (R₀, R₁), R₃

Inner Comparebyte R₃, (R₀, R₂)



Branch <= 0 Next

Movebyte (R₀, R₂), R₄

Movebyte R₃, (R₀, R₂)

Movebyte R₄, (R₀, R₁)

Next Decrement R2

Branch T=0 Inner

Decrement R1

Branch T=0 ~~Reset~~ out

فعندها لو نتيجة
المقارنة أكبر من المفتر
صريج نزول ال (loop)
(التي اسمها (Inner))

4)

ORIGIN 1000

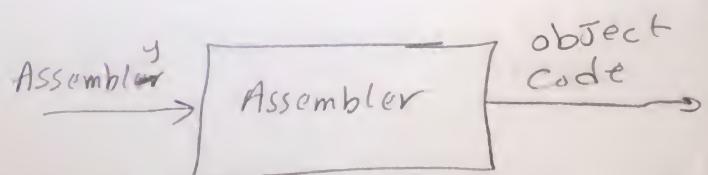
DATAWORD 300

حالاته بستة معايير

الخطية لكنه ارثان مختل

Assembler Directives الأدلة

MOVE #300, 1000



.Code ينفذ ال

- execute (Assembler) (عثنا بـ execute (Assembler))

4)

5

عملية (Subroutine) البرمجة (Function)

Move ~~#~~ AVEC, R1

Move ~~#~~ BVEC, R2

Move N, R3

Call Sub

Move ~~R0~~, DOT PROD

Sub

CLR R0

loop

Move (R1)+, R4

Multiply (R2)+, R4

ADD R4, R0

Decrement R3

Branch to loop

Return

loop (loop) تكرار

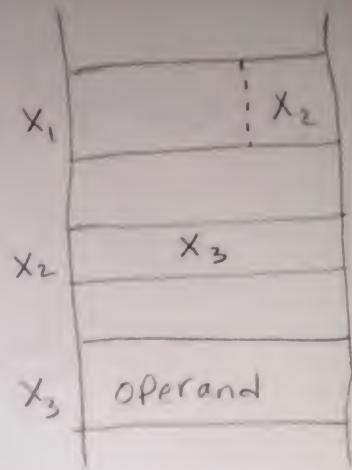
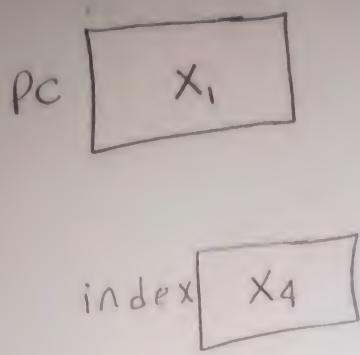
clr (CLR) نص

mult (الناتج الى R0) نص

-R0 كتباه في ~~R0~~

5

6



الحالات الاربعة ←

Direct $\Rightarrow x_3 = x_2$

Indirect $\Rightarrow x_3 = (x_2)$

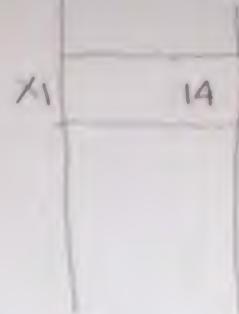
PC relative $\Rightarrow x_3 = x_2 + x_1 + 1$ 1 يزيد (PC)
عندما يتغير على الامر
اللذالى.

Indexed $\Rightarrow x_3 = x_2 + x_4$

6

7

--- ال تخمير في ال درجة
عوان
(oprand) ---



immediate

oprand is 14

move #14, R1

--- جملة R1 #14

move 14, R1

--- جملة R1 14

R1 is 14

Direct

⇒ 14

in memory location whose address
is 14

Indirect

register

register --- عوان 14

--- جملة move 14, register

register indirect

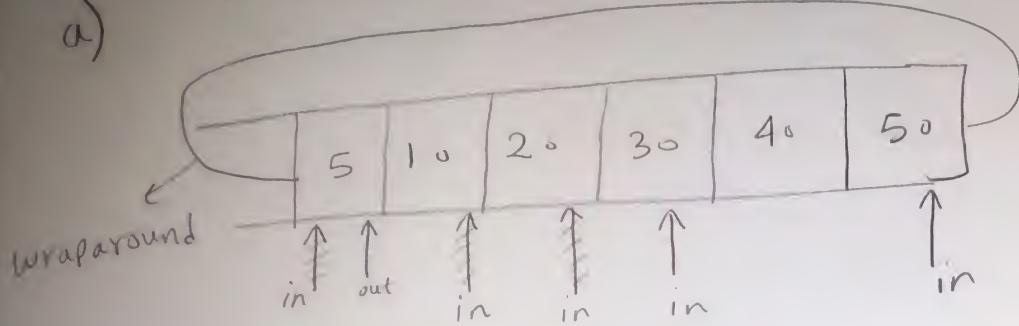
memory loc. --- whose address is
register 14.

7

8

a)

FIFO



- Queue \rightarrow (in) يستمر في جزو 11 ←

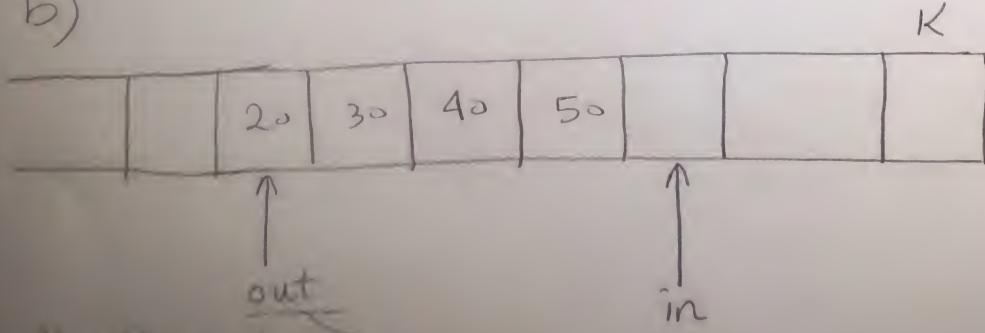
~ 5 \rightarrow 10 \rightarrow 15 \rightarrow 16 (out) ←
ـ 5 \rightarrow 10 \rightarrow 15 \rightarrow 16 \rightarrow Queue 11

\rightarrow سبط نقط لعدة يدخل (50) ويغادر معاً ←

Wraparound يغادر بعدل

(a) حل السؤال

b)



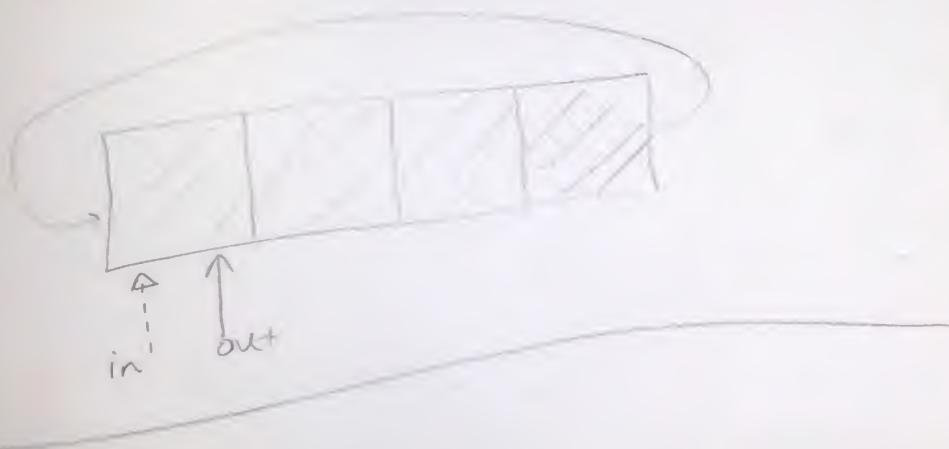
ـ سبط نقط

ـ 100

8

الحل

ج) يُعَدُّ داعمًا لـ $(In = out)$ * عندما تكون
 ومتى $(Queue)$ فما \sim
 in {ستُرِّي} ملء الأماكن الفارغة وعمل (in) *
 ووجد \sim ممكناً فـ in يقوم بـ $pushing$.



ج) $(in + 1) \bmod K$ \rightarrow K قيم



$in = 2, K = 5$

طلما القيمة أقل من K
 الناتج يساوي لها.

$4 \bmod 5 = \boxed{4}$

ج)

d)

$Loc \leftarrow in$

$out = \leftarrow (int + 1) \bmod K$

(operation) \rightarrow (out) \leftarrow
 (out) \leftarrow (int $\bmod K$)

9

Report

10

sheet 4

WMFC

1 - PCout, MARin, Read, select 9, Add, Zin

2 - Zout, PCin, Xin, WMFC

3 - MDRout, IRin

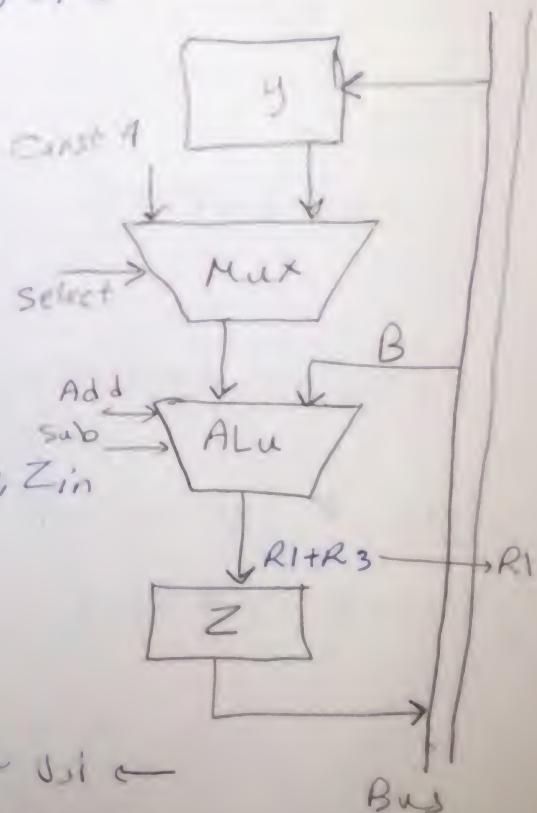
4 - R3out, MARin, Read

5 - ~~R1~~out, Xin, WMFC

6 - MDRout, select 9, ADD, Zin

7 - Zout, R1in, End

(Fetch) الـ 32 دايمز هما أوامر الـ
فاتسني 3 كل البرامج.



Q ADD (R3), R1

هو (indirect) بسيط خطوة رقم (4) لاتس
(MAR) خرجنا منه اه

7 clock cycles خطا فاذا هيأخذوا ←
execution time

3 (clockcycles) يأخذ عدد ← WMC

∴ execution time = 9 clock cycle

4 delay \Rightarrow Bus = 0.3 ns, ALU = 2 ns
setup \rightarrow 0.2 ns, Hold time \rightarrow 0

أقل وقت تأخير ممكن

time = 2.5 ns